

Self Assessment

Criterion ↓	Amazing (A)	Good (B)	OK (C+/C-/C-)	Not There Yet (I)
Score →	40-----35	34-----29	28-----20	19-----0
Originality	What a great concept! This is nothing like any of the examples on KhanAcademy, and it's not like anything else I've seen before, either...	I can see that some ideas came from elsewhere, but they are put together in new and interesting ways.	Oh, look! It's another "Angry Birds" clone, only with dogs instead of birds... not much creativity is evident.	Um, you copied the program and changed one or two minor bits. Where's YOUR idea and skill?
Design	This game is easy on the eyes and uses the principles and elements of design to make it appealing to everyone. Great job!	It's pretty professional-looking, for a quick Java game, but some minor tweaks are necessary to make this really stellar.	Yeah...you could have chosen colors, shapes and/or other elements better. Not enough time was spent on the game's appearance.	What is this mess? Looks extremely amateurish. Elements and principles of design clearly not considered.
Technical Skills	Not only does the programmer show off the topics from the tutorials, but also shows that there has been some extra research on the side to learn some new tricks.	Everything from the tutorials is there. Nice.	Almost everything from the tutorials is present in the game. One or two points may not be covered.	It's hard to tell from playing your game and reading the code that you really understand much of anything you were supposed to learn.
Instructions	Your game has clear instructions; the players can figure out what they are supposed to do in order to win without having to actually play the game. No spelling or grammatical errors in your instructions.	Your game's instructions are mostly clear. There may be spelling/grammatical errors, but they don't interfere with the meaning.	Your players have to actually play the game to figure out how it works, even though you included directions. You may have errors in the instructions that leave the players scratching their heads.	You did not include instructions for how to play (0); or even after playing the game several times, your teacher can't figure out how a person can win (or lose) the game.
Code & Documentation	Your code is elegant and uses appropriate shortcuts (if else instead of two or three nested if-loops, for example). Documentation is clear: just by looking at your code, we can see what each part is for. Tabs are used to indent nested functions.	You use shortcuts sometimes, but not others. Documentation is in fragments, or not completely clear. Tabs may be used sometimes and not other times.	You have a command of the basics, but shortcuts, tabs, and documentation make other programmers say, "Whuuuuut?" as they read. They can figure it out...but it takes time and effort.	Completely inadequate documentation and formatting makes your program (almost) impossible to deconstruct.

